Assignment 7: Time Series Analysis

Aubrey Knier

OVERVIEW

This exercise accompanies the lessons in Environmental Data Analytics on time series analysis.

Directions

1. Change "Student Name" on line 3 (above) with your name.
2. Work through the steps, **creating code and output** that fulfill each instruction.
3. Be sure to **answer the questions** in this assignment document.
4. When you have completed the assignment, **Knit** the text and code into a single PDF file.
5. After Knitting, submit the completed exercise (PDF file) to the dropbox in Sakai. Add your last name into the file name (e.g., "Fay_A07_TimeSeries.Rmd") prior to submission.

The completed exercise is due on Monday, March 14 at 7:00 pm.

Set up

1. Set up your session:

- Check your working directory
- Load the tidyverse, lubridate, zoo, and trend packages
- Set your ggplot theme

```
#1
getwd()
```

```
## [1] "/Users/aubreyknier/Desktop/Spring 2022/ENV872_EDA/Environmental_Data_Analytics_2022"
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(trend)

mytheme <- theme_classic(base_size = 14) +
  theme(axis.text = element_text(color = "sienna3"),
        legend.position = "right")
theme_set(mytheme)
```

2. Import the ten datasets from the Ozone_TimeSeries folder in the Raw data folder. These contain ozone concentrations at Garinger High School in North Carolina from 2010-2019 (the EPA air database only allows downloads for one year at a time). Import these either individually or in bulk and then combine them into a single dataframe named `GaringerOzone` of 3589 observation and 20 variables.

```
#2
O3_2010_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2010_raw.csv",
                         stringsAsFactors = T)
O3_2011_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2011_raw.csv",
                         stringsAsFactors = T)
O3_2012_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2012_raw.csv",
                         stringsAsFactors = T)
O3_2013_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2013_raw.csv",
                         stringsAsFactors = T)
O3_2014_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2014_raw.csv",
                         stringsAsFactors = T)
O3_2015_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2015_raw.csv",
                         stringsAsFactors = T)
O3_2016_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2016_raw.csv",
                         stringsAsFactors = T)
O3_2017_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2017_raw.csv",
                         stringsAsFactors = T)
O3_2018_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2018_raw.csv",
                         stringsAsFactors = T)
O3_2019_data <- read.csv("./Data/Raw/Ozone_TimeSeries/EPAair_O3_GaringerNC2019_raw.csv",
```

```
                        stringsAsFactors = T)

O3_total_data <- rbind(O3_2010_data, O3_2011_data, O3_2012_data, O3_2013_data, O3_2014_data,
                       O3_2015_data, O3_2016_data, O3_2017_data,
                       O3_2018_data, O3_2019_data)

dim(O3_total_data)
```

```
## [1] 3589   20
```

Wrangle

3. Set your date column as a date class.

4. Wrangle your dataset so that it only contains the columns Date, Daily.Max.8.hour.Ozone.Concentration, and DAILY_AQI_VALUE.

5. Notice there are a few days in each year that are missing ozone concentrations. We want to generate a daily dataset, so we will need to fill in any missing days with NA. Create a new data frame that contains a sequence of dates from 2010-01-01 to 2019-12-31 (hint: `as.data.frame(seq())`). Call this new data frame Days. Rename the column name in Days to "Date".

6. Use a `left_join` to combine the data frames. Specify the correct order of data frames within this function so that the final dimensions are 3652 rows and 3 columns. Call your combined data frame GaringerOzone.

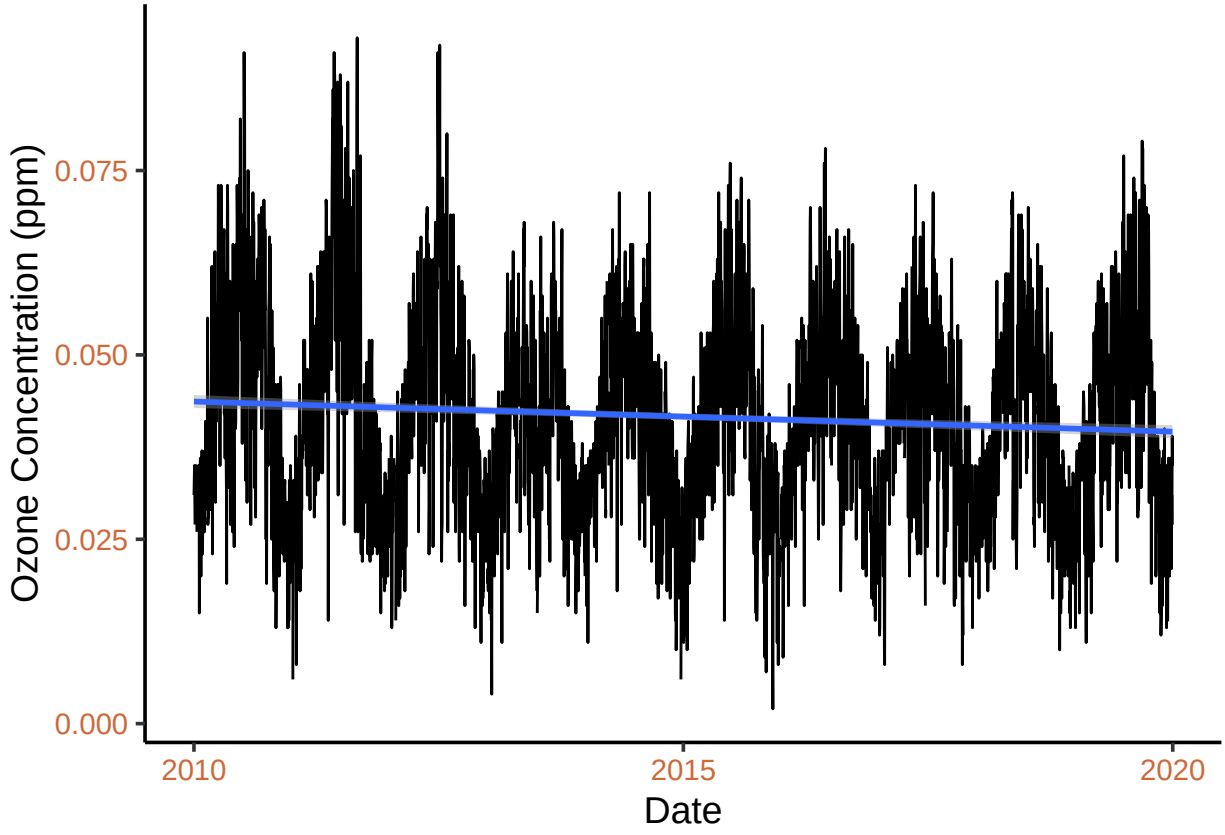
```
# 3
O3_total_data$Date <- as.Date(O3_total_data$Date, format="%m/%d/%Y")

# 4
O3_total_data_subset <- O3_total_data %>%
  select(Date, Daily.Max.8.hour.Ozone.Concentration, DAILY_AQI_VALUE)

# 5
Days <- as.data.frame(seq(as.Date("2010-01-01"), to=as.Date("2019-12-31"), by="days"))
colnames(Days) <- "Date"

# 6
GaringerOzone <- left_join(Days, O3_total_data_subset, by="Date")
```

Visualize

7. Create a line plot depicting ozone concentrations over time. In this case, we will plot actual concentrations in ppm, not AQI values. Format your axes accordingly. Add a smoothed line showing any linear trend of your data. Does your plot suggest a trend in ozone concentration over time?

```
#7
ggplot(GaringerOzone, aes(x=Date, y=Daily.Max.8.hour.Ozone.Concentration)) +
    geom_line() +
      geom_smooth(method=lm) +
        ylab("Ozone Concentration (ppm)")
```

```
## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 63 rows containing non-finite values (stat_smooth).
```



Answer: The trend line of my plot suggests that there is an overall slight decreasing trend in ozone concentration over time. There also seems to be a seasonal cycle across the data.

Time Series Analysis

Study question: Have ozone concentrations changed over the 2010s at this station?

8. Use a linear interpolation to fill in missing daily data for ozone concentration. Why didn't we use a piecewise constant or spline interpolation?

```
#8
GaringerOzone_filled <-
  GaringerOzone %>%
  mutate(ppm.filled = zoo::na.approx(Daily.Max.8.hour.Ozone.Concentration))

summary(GaringerOzone_filled$Daily.Max.8.hour.Ozone.Concentration) #63 NAs
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 0.00200 0.03200 0.04100 0.04163 0.05100 0.09300      63
```

```
summary(GaringerOzone_filled$ppm.filled) #no NAs
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00200 0.03200 0.04100 0.04151 0.05100 0.09300
```

Answer: We did not use a piecewise constant interpolation to fill in missing daily data for ozone concentrations because the assumption that any missing data point is equal to the data points next to it would be inappropriate since data measurements seem to increase or decrease from one point to the next in seasonal cycles. We did not use a spline interpolation because we have a linear (straight line) trend, and spline interpolation uses a quadratic function.

9. Create a new data frame called `GaringerOzone.monthly` that contains aggregated data: mean ozone concentrations for each month. In your pipe, you will need to first add columns for year and month to form the groupings. In a separate line of code, create a new Date column with each month-year combination being set as the first day of the month (this is for graphing purposes only)

```
#9
GaringerOzone.monthly <- GaringerOzone_filled %>%
  mutate("Month"=month(Date), "Year"=year(Date)) %>%
    group_by(Year, Month) %>%
      summarize("Monthly Mean" = mean(ppm.filled))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

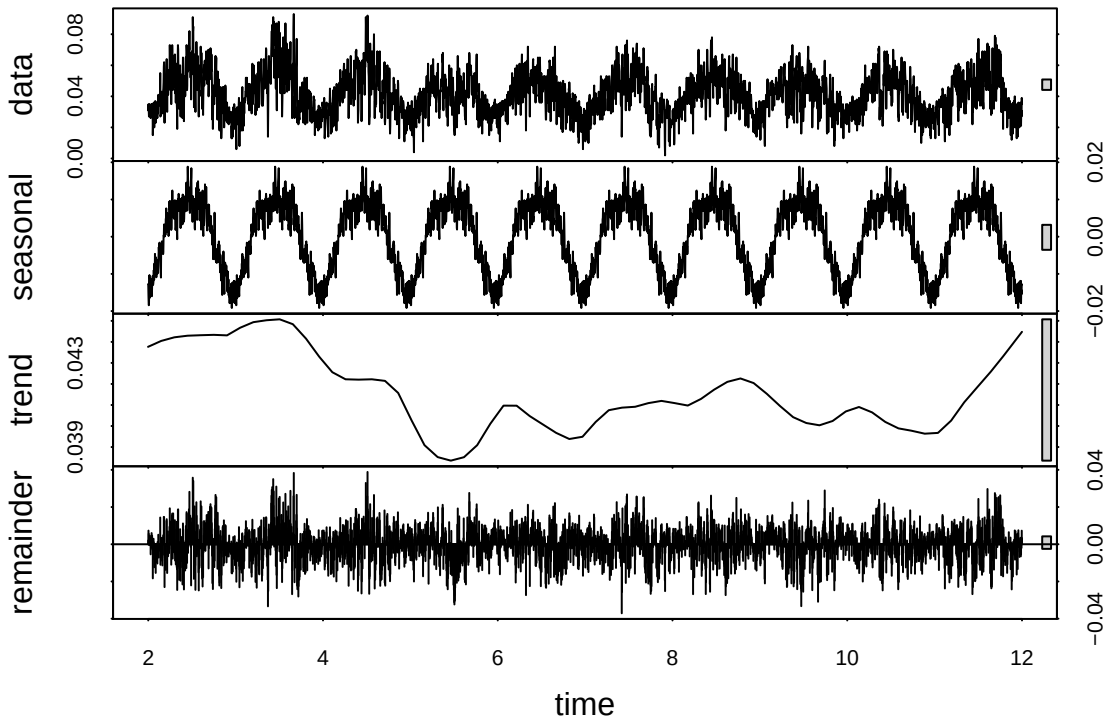
```
GaringerOzone.monthly$Date <- seq(as.Date("2010-01-01"), to=as.Date("2019-12-31"), by="months")
```

10. Generate two time series objects. Name the first `GaringerOzone.daily.ts` and base it on the dataframe of daily observations. Name the second `GaringerOzone.monthly.ts` and base it on the monthly average ozone values. Be sure that each specifies the correct start and end dates and the frequency of the time series.

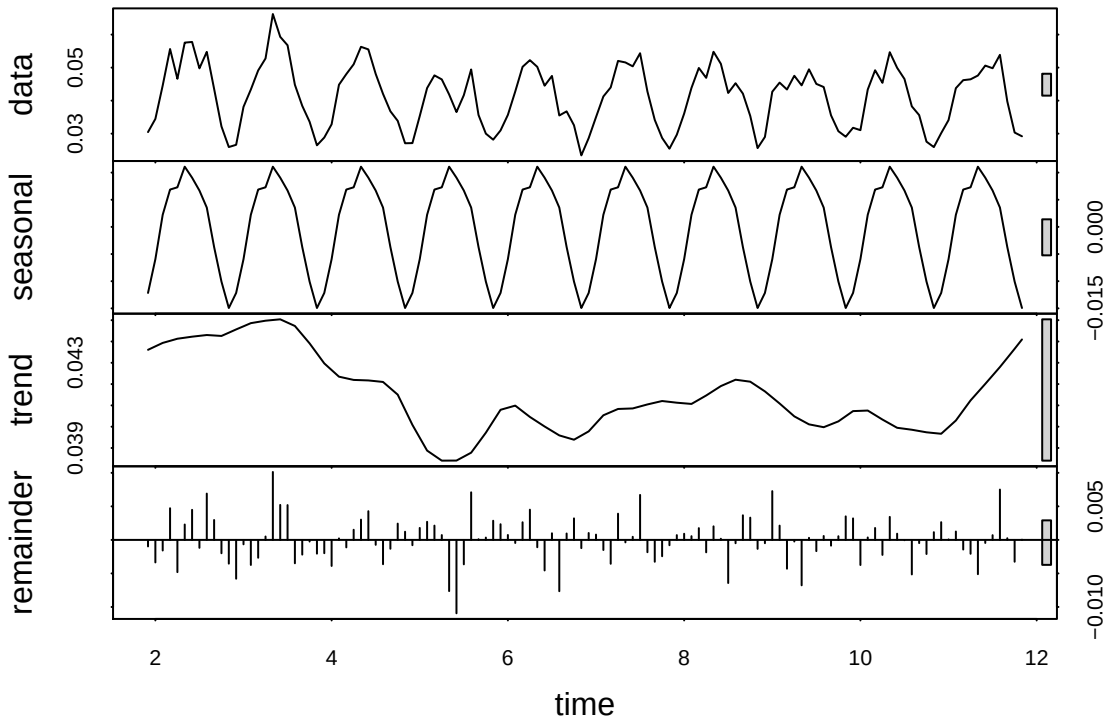
```
#10
GaringerOzone.daily.ts <- ts(GaringerOzone_filled$ppm.filled, start(2010, 1),
                             frequency=365) #end date specified automatically

GaringerOzone.monthly.ts <- ts(GaringerOzone.monthly$`Monthly Mean`,
                               start(2010, 1),
                               frequency=12) #end date specified automatically
```

11. Decompose the daily and the monthly time series objects and plot the components using the `plot()` function.

```
#11
GaringerOzone.daily.ts.decomposed <- stl(GaringerOzone.daily.ts, s.window = "periodic")
plot(GaringerOzone.daily.ts.decomposed)
```




```
GaringerOzone.monthly.ts.decomposed <- stl(GaringerOzone.monthly.ts, s.window = "periodic")
plot(GaringerOzone.monthly.ts.decomposed)
```



12. Run a monotonic trend analysis for the monthly Ozone series. In this case the seasonal Mann-Kendall is most appropriate; why is this?

```
#12
Ozone_monthly_trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.ts)
Ozone_monthly_trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(Ozone_monthly_trend) #p=0.046724, data has trend
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

```
Ozone_monthly_trend2 <- trend::smk.test(GaringerOzone.monthly.ts)
Ozone_monthly_trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.ts
## z = -1.963, p-value = 0.04965
```

```
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

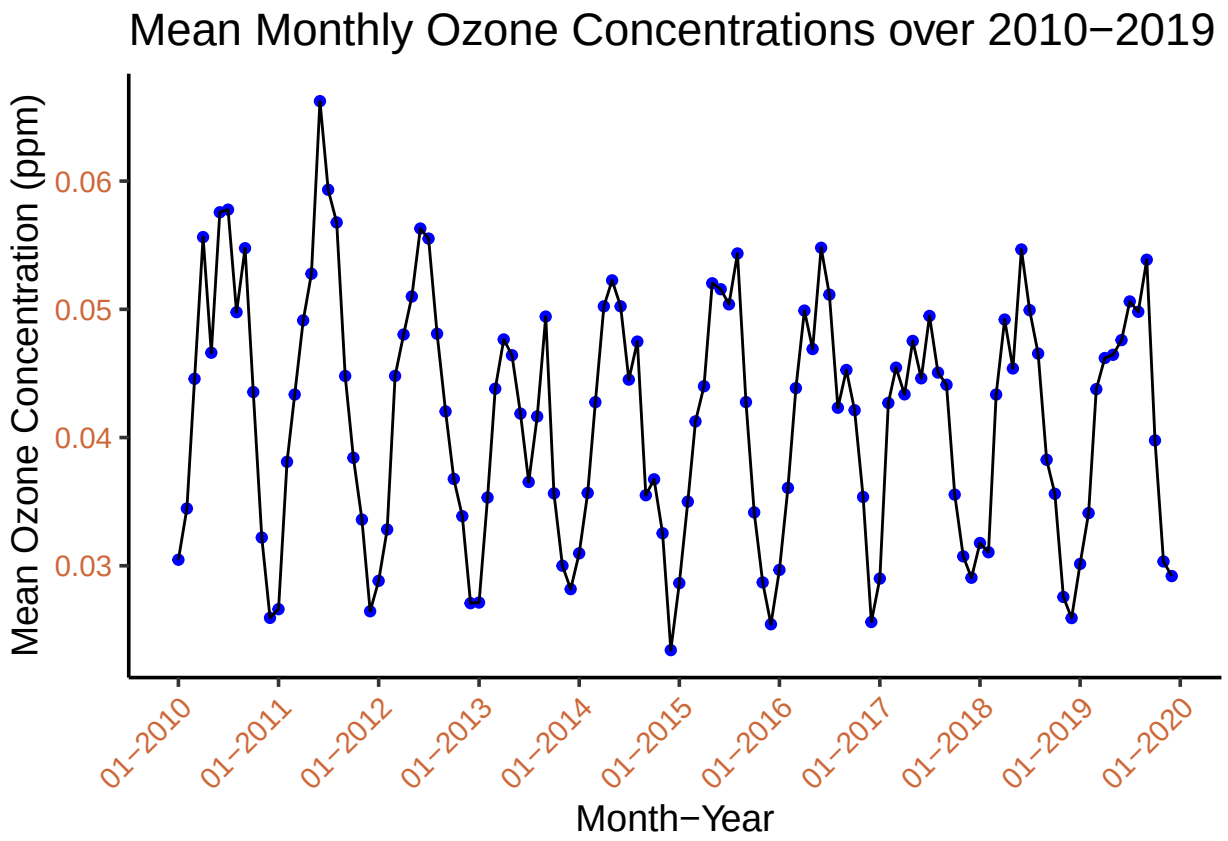
```
summary(Ozone_monthly_trend2) #no individual seasons with p<0.05
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 2:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 3:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 4:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 5:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 6:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 7:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 8:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 9:   S = 0  -13  125 -0.289 -1.073  0.28313
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0  11  125  0.244  0.894  0.37109
## Season 12:   S = 0  15  125  0.333  1.252  0.21050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: The seasonal Mann-Kendall monotonic trend analysis is most appropriate for our data because seasonality is present (as seen in the plots of our decomposed time series components), and this is the only analysis that can handle seasonal data. This test reveals if/how ozone concentration changes over the years 2010-2019 when the seasonal component is incorporated.

13. Create a plot depicting mean monthly ozone concentrations over time, with both a geom_point and a geom_line layer. Edit your axis labels accordingly.

```
# 13
Ozone_monthly_plot <- ggplot(GaringerOzone.monthly,
                             aes(x=Date, y=`Monthly Mean`)) +
    geom_point(col="blue") +
      geom_line() +
        ylab("Mean Ozone Concentration (ppm)") +
          xlab("Month-Year") +
          scale_x_date(date_breaks = "years", date_labels="%m-%Y") +
  theme(axis.text.x = element_text(angle = 45, hjust=1)) +
    ggtitle("Mean Monthly Ozone Concentrations over 2010-2019")

print(Ozone_monthly_plot)
```



14. To accompany your graph, summarize your results in context of the research question. Include output from the statistical test in parentheses at the end of your sentence. Feel free to use multiple sentences in your interpretation.

    Answer: The results of this time series analysis conclude that ozone concentrations have changed over the 2010s at this station. A seasonal Mann-Kendall trend test indicated that a significant overall trend is present (p=0.047), while no individual seasons of the month exhibited a significant trend individually (p>0.15 in each season). These results conclude that the ozone concentration has declined over the years 2010-2019, regardless of changes due to seasonality.

15. Subtract the seasonal component from the `GaringerOzone.monthly.ts`. Hint: Look at how we extracted the series components for the EnoDischarge on the lesson Rmd file.

16. Run the Mann Kendall test on the non-seasonal Ozone monthly series. Compare the results with the ones obtained with the Seasonal Mann Kendall on the complete series.

```
#15
Ozone.Components <- as.data.frame(GaringerOzone.monthly.ts.decomposed$time.series[,1:3])

GaringerOzone.monthly.non.seasonal.ts <- GaringerOzone.monthly.ts - Ozone.Components$seasonal

#16
Ozone.non.seasonal.monthly.trend <- Kendall::SeasonalMannKendall(GaringerOzone.monthly.non.seasonal.ts)
Ozone.non.seasonal.monthly.trend
```

```
## tau = -0.143, 2-sided pvalue =0.046724
```

```
summary(Ozone.non.seasonal.monthly.trend) #p=0.046724
```

```
## Score =  -77 , Var(Score) = 1499
## denominator =  539.4972
## tau = -0.143, 2-sided pvalue =0.046724
```

```
Ozone.non.seasonal.monthly.trend2 <- trend::smk.test(GaringerOzone.monthly.non.seasonal.ts)
Ozone.non.seasonal.monthly.trend2
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data:  GaringerOzone.monthly.non.seasonal.ts
## z = -1.963, p-value = 0.04965
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##    S varS
##  -77 1499
```

```
summary(Ozone.non.seasonal.monthly.trend2)
```

```
##
##  Seasonal Mann-Kendall trend test (Hirsch-Slack test)
##
## data: GaringerOzone.monthly.non.seasonal.ts
## alternative hypothesis: two.sided
##
## Statistics for individual seasons
##
## H0
##                       S varS    tau      z Pr(>|z|)
## Season 1:   S = 0   -1  125 -0.022  0.000  1.00000
## Season 2:   S = 0   -4  124 -0.090 -0.269  0.78762
## Season 3:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 4:   S = 0  -15  125 -0.333 -1.252  0.21050
## Season 5:   S = 0  -17  125 -0.378 -1.431  0.15241
## Season 6:   S = 0  -11  125 -0.244 -0.894  0.37109
## Season 7:   S = 0   -7  125 -0.156 -0.537  0.59151
## Season 8:   S = 0   -5  125 -0.111 -0.358  0.72051
## Season 9:   S = 0  -13  125 -0.289 -1.073  0.28313
## Season 10:   S = 0 -13  125 -0.289 -1.073  0.28313
## Season 11:   S = 0  11  125  0.244  0.894  0.37109
## Season 12:   S = 0  15  125  0.333  1.252  0.21050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer: When the seasonal component of the time series is removed, the seasonal Mann-Kendall test provides the same result as when run with the complete series (p=0.047). This is because this test already incorporates seasonality in the analysis, so it made no difference to leave in or remove the seasonal component of the time series. Comparing the test's results with and without the seasonal component demonstrates this.